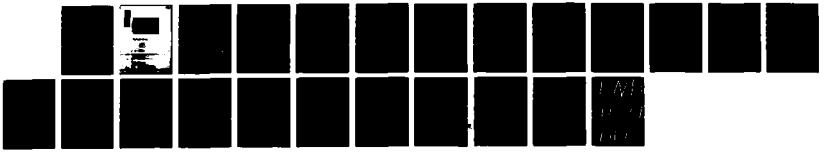
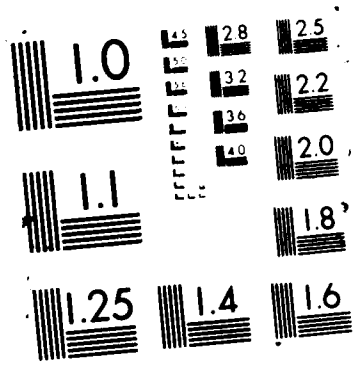
AD-A186 148 A MULTICUT ALGORITHM FOR TWO-STAGE STOCHASTIC LINEAR PROGRAMS(U) MICHIGAN UNIV ANN ARBOR DEPT OF INDUSTRIAL AND OPERATIONS ENGINEERING J R BIRGE ET AL. DEC 86 1/1

UNCLASSIFIED TR-85-15 N00014-86-K-0628 F/G 12/4 NL




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8

A Multicut Algorithm for Two-Stage Stochastic Linear Programs

John R. Birge
Dept. of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan, USA 48109

Francois V. Louveaux
Faculte des Sciences Economiques et Sociales
Facultes Universitaires N-D de la Paix
B-5000

Belgium

Technical Report 85-15

Department of Industrial and Operations Engineering

(12)

A Multicut Algorithm for Two-Stage Stochastic Linear Programs

John R. Birge
Dept. of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan, USA 48109

Francois V. Louveaux
Faculte des Sciences Economiques et Sociales
Facultes Universitaires N-D de la Paix
B-5000

Belgium

Technical Report 85-15

Revised December 1986

A Multicut Algorithm for Two–Stage Stochastic Linear Programs

John R. Birge*
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan, USA 48109

François V. Louveaux
Faculté des Sciences Economiques et Sociales
Facultés Universitaires N–D de la Paix
B-5000
Belgium

Abstract : Outer linearisation methods, such as Van Slyke and Wets's L–shaped method for stochastic linear programs, generally apply a single cut on the nonlinear objective at each major iteration. The structure of stochastic programs allows for several cuts to be placed at once. This paper describes a multicut algorithm to carry out this procedure. It presents experimental and theoretical justification for reductions in major iterations.

Keywords: stochastic programming, outer linearization, cutting plane methods.

* - This author's work was supported in part by National Science Foundation Grant No. ECS–8304065, Office of Naval Research Grant N00014-86-K-0628, by Dalhousie University during a visit in the Department of Mathematics, Statistics, and Computing Science, and by the National Research Council under a Research Associateship at the Naval Postgraduate School, Monterey, California.

1. Introduction

Two-stage stochastic linear programs have a deterministic equivalent program with convex objective function that can be solved by a variety of methods. The L-shaped method of Van Slyke and Wets [12] is a cutting plane or outer linearisation technique for solving this program when the random variables have finite support. It has been extended to multi-stage stochastic linear and quadratic programs by Birge [3] and Louveaux [10], respectively. Their analyses showed the L-shaped algorithm to be an effective solution technique for a variety of examples. The structure of stochastic programs, however, allows the L-shaped method to be extended to include multiple cuts on the objective in each major iteration. This paper describes this procedure for two–stage stochastic linear programs. A multi–stage version has been proposed by Silverman [11].

Adding multiple cuts at each iteration of an outer linearisation procedure corresponds in the dual to including several columns in the master problem of an inner linearisation algorithm such as Dantzig–Wolfe decomposition (see, for example, Lasdon [9]). In inner linearisation, adding several columns instead of a single aggregated column may speed up convergence (see Birge [2]) and reduce the number of major iterations. The same types of behavior may arise in outer linearisation, but only qualitative descriptions have been given. In this paper, we quantify this phenomenon by using the problem structure to derive worst–case bounds on the number of major iterations in the single and multiple cut cases. These results are supported by experiments on practical test problems.

In Section 2, we briefly describe the L–shaped algorithm and the problem structure. In Section 3, we present the multicut algorithm and, in Section 4, we discuss its efficiency in terms of bounds on the number of major iterations for general problems. The specific case of simple recourse problems is discussed in Section 5. Section 6 presents results of numerical experiments and the appendices provide illustrative examples of claims made in the text.

2. The L–shaped algorithm

The classical two-stage stochastic linear program with fixed recourse is the problem of finding

$$\begin{array}{lllll} \min & z = cx & +\mathrm{E}_\xi[\min q(\omega)y(\omega)] & & \\ \text{s.t.} & Ax & & = b & \\ & T(\omega)x & +Wy(\omega) & = h(\omega) & \\ & x \geq 0 & y(\omega) \geq 0 \text{ a.s.} & & \end{array} \tag{1}$$

where c is a known vector in $\Re^{n_1}$, b a known vector in $\Re^{m_1}$, ξ is a random N–vector defined on the probability space, (Ω, Ξ, P), and A and W are known matrices of sizes $m_1 \times n_1$ and $m_2 \times n_2$, respectively. W is called the *recourse matrix*.

For each ω, $T(\omega)$ is $m_2 \times n_1$, $q(\omega) \in \Re^{n_2}$ and $h(\omega) \in \Re$. Piecing together the stochastic components of the problem, we obtain a vector $\xi(\omega) = (q(\omega), h(\omega), T_1(\omega), ..., T_{m_2}(\omega))$ with $N = n_2 + m_2 + (n_2 \times n_1)$ components, where $T_i(\omega)$ is the i*th* row of $T(\omega)$. Transposes have been eliminated for simplicity. E_ξ represents the mathematical expectation with respect to ξ.

A precise formulation of (1) is given by the *deterministic equivalent program (D.E.P.)*:

$$\begin{array}{lll} \min & z = cx & +\mathcal{Q}(x) \\ \text{s.t.} & Ax & = b \\ & x & \geq 0, \end{array} \tag{2}$$

where

$$\mathcal{Q}(x) = \mathrm{E}_\xi Q(x, \xi(\omega)),$$

and

$$Q(x, \xi(\omega)) = \min\{q(\omega)y(\omega) \mid Wy(\omega) = h(\omega) - T(\omega)x, y \geq 0\}.$$

Properties of the D.E.P. have been extensively studied (Wets [13], Garstka and Wets [6]). Of particular interest for computational aspects is the fact that $Q(x, \xi)$ is a convex piecewise linear function of x and that $\mathcal{Q}(x)$ is also piecewise linear convex if ξ has finite support.

When T is non–stochastic, the original formulation (2) can be replaced by

$$\begin{array}{lrll} \min & z = cx & +\Psi(\chi) & \\ \text{s.t.} & Ax & & = b \\ & Tx & -\chi & = 0 \\ & x & \geq 0 & \end{array} \tag{3}$$

where $\Psi(\chi) = \mathrm{E}_\xi \psi(\chi, \xi(\omega))$ and $\psi(\chi, \xi(\omega)) = \min\{q(\omega)y(\omega) \mid Wy(\omega) = h(\omega) - \chi, y \geq 0\}$. This formulation stresses the fact that choosing x corresponds to generating an m_2–dimensional tender $\chi = Tx$ to be "bid" against the outcomes $h(\omega)$ of the random events.

In this paper, we concentrate on algorithms for solving (2) or (3). Excluding algorithms for the specific simple recourse problem (see, e.g., Kall [8], Wets [14]), a major method for solving (2) is the L–shaped algorithm due to Van Slyke and Wets [12] which is an outer linearisation procedure as in Benders' decomposition (Benders [1]). For more details on other algorithmic procedures, see the discussion in Wets [14]. Outer linearisation is generally preferred to inner linearisation of the dual because the dual generally has more rows than the primal and, hence, requires more work per iteration. Outer linearisation is also generally preferred to basis factorisation in stochastic linear programming because basis requires storing a basis for each realisation of ξ. In outer linearisation, these bases need not be stored. Efficient procedures (see Wets [14]) may then be used to solve $Q(x, \xi(\omega))$ for large numbers of realisations of ξ.

The L-shaped method consists of solving an approximation of (2) by using an outer linearization of $\mathcal{Q}$. Two types of constraints are sequentially added: (i) feasibility cuts (5) determining $\{x|\mathcal{Q}(x) < +\infty\}$ and (ii) optimality cuts (6) which are linear approximations to $\mathcal{Q}$ on its domain of finiteness.

Assumption: The random variable ξ has finite support.

Let $k = 1, ..., K$ index the possible realisations of ξ with probabilities p_k.

L–shaped algorithm

Step 0. Set $s = t = \nu = 0$.

Step 1. Set $\nu = \nu + 1$. Solve the linear program (4) - (6).

$$\begin{array}{lllll} \min & z = cx & +\theta & & (4) \\ \text{s.t.} & Ax & & = b & \\ & D_l x & & \geq d_l, l = 1, ..., s & (5) \\ & E_l x & +\theta & \geq e_l, l = 1, ..., t, & (6) \\ & x \geq 0, & \theta \in \Re. & & \end{array}$$

Let (x^ν, θ^ν) be an optimal solution. If no constraint (6) is present, θ is set equal to $-\infty$ and is ignored in the computation.

Step 2. For $k = 1, ..., K$ solve the linear program

$$\begin{array}{llllll} \min & w^1 = & ev^+ & +ev^- & & \\ \text{s.t.} & Wy & +Iv^+ & -Iv^- & & = h_k - T_k x^\nu \\ & y \geq 0, & v^+ \geq 0, & v^- \geq 0, & & \end{array}$$

where $e = (1, ..., 1)$, until, for some k, the optimal value $w^1 > 0$. Let σ^ν be the associated simplex multipliers and define

$$D_{s+1} = \sigma^\nu T_k$$

and

$$d_{s+1} = \sigma^\nu h_k$$

to generate a feasibility cut of type (5). Set $s = s + 1$ and return to Step 1. If, for all k, $w^1 = 0$, go to Step 3.

Step 3. For $k = 1, ..., K$ solve the linear program

$$\begin{array}{llll} \min w^2 = & q_k y & & \\ \text{s.t.} & Wy & = h_k - T_k x^\nu & \\ & y & \geq 0. & \end{array} \qquad (7)$$

Let π_k^ν be the simplex multipliers associated with the optimal solution of Problem k of type (7). Define

$$E_{t+1} = \sum_{k=1}^{K} p_k \pi_k^\nu T_k \qquad (8)$$

and

$$e_{t+1} = \sum_{k=1}^{K} p_k \pi_k^\nu h_k. \tag{9}$$

Let $w^{2\nu} = e_{t+1} - E_{t+1}x^\nu$. If $\theta^\nu \geq w^{2\nu}$, stop, x^ν is an optimal solution. Otherwise, set $t = t + 1$, and return to Step 1.

Improvements in this algorithm have been given in two directions: (i) the study of cases in which Step 2 can be modified to solve only one linear program instead of N and (ii) the study of bunching and sifting procedures to reduce the work in Step 3 (Garstka and Rutenberg [5]). We again refer to Wets [14] for a detailed account of these improvements.

In this paper, we propose to replace the outer linearisation of $\mathcal{Q}$ used in the L-shaped method by an outer linearisation of all functions

$$Q_k(x) = \min\{q_k y \mid Wy = h_k - T_k x, y \geq 0\}, \tag{10}$$

of which $\mathcal{Q}(x)$ constitutes the expectation, i.e. $\mathcal{Q}(x) = \sum_{k=1}^{K} p_k Q_k(x)$.

3. The Multicut L-shaped Algorithm

The multicut L-shaped algorithm is defined as follows:

Step 0. Set $s = \nu = 0$ and $t_k = 0$ for all $k = 1, \ldots, K$.

Step 1. Set $\nu = \nu + 1$. Solve the linear program (11) - (13).

$$\begin{aligned}
\min z = \quad & cx + \sum_{k=1}^{K} \theta_k && && (11)\\
\text{s.t.} \quad & Ax && = b &\\
& D_l x && \geq d_l, l = 1, \ldots, s && (12)\\
& E_{l(k)} x + \theta_k && \geq e_{l(k)}, l(k) = 1, \ldots, t(k), k = 1, \ldots, K, && (13)\\
& x && \geq 0,
\end{aligned}$$

Let $(x^\nu, \theta_1^\nu, \ldots, \theta_K^\nu)$ be an optimal solution of (11). If no constraint (13) is present for some k, θ_k^ν is set equal to $-\infty$ and is ignored in the computation.

Step 2. As before.

Step 3. For $k = 1, ..., K$ solve the linear program (7).

Let π_k^ν be the simplex multipliers associated with the optimal solution of problem k. If

$$\theta_k^\nu < p_k \pi_k^\nu (h_k - T_k x^\nu) \tag{14}$$

define

$$E_{t(k)+1} = p_k \pi_k^\nu T_k \tag{15}$$

and

$$e_{t(k)+1} = p_k \pi_k^\nu h_k \tag{16}$$

and set $t(k) = t(k) + 1$.

If (14) does not hold for any $k = 1, ..., K$, stop, x^ν is an optimal solution. Otherwise, return to Step 1.

We illustrate the differences and similarities between the multicut approach and the standard L–shaped algorithm in the example of Appendix A. The multicut approach is based on the idea that using outer approximations of all $Q_k(x)$ sends more information than a single cut on $Q(x)$ and that, therefore, fewer iterations are needed.

4. Efficiency and bounds

The following dominance property can be established. Define $K_1 \cap K_2$ to be the constraint set of the stochastic program (2), where

$$K_1 = \{x | Ax = b, x \geq 0\}$$

and

$$K_2 = \cap_{\xi \in \Xi} \{x \mid \exists y \geq 0 \text{ such that } Wy = h(\xi) - T(\xi)x\},$$

where by assumption Ξ is finite.

$\mathcal{Q}(x)$ is known to be piecewise linear, hence there exists a polyhedral decomposition of $K_1 \cap K_2$ into a finite collection of closed convex sets C_r, called the *cells* of the decomposition, such that the intersection of two distinct cells has an empty interior and such that the function $\mathcal{Q}(x)$ is either identically $-\infty$ or affine on each cell.

The L–shaped method outerlinearises $\mathcal{Q}(x)$ by identifying a *facet*, the function graph for one cell of the decomposition, on each return to Step 1. The multicut algorithm outerlinearizes $Q_k(x)$. On each return to Step 1, the multicut algorithm identifies a facet of some $Q_k(x)$. This information may, however, be equivalent to identifying several facets of $\mathcal{Q}$ since each *combination* of facets of Q_k corresponds to a *single* facet of $\mathcal{Q}$. This property enables the multicut algorithm to converge faster than the L–shaped algorithm.

In the following proposition, we state this more precisely. We define a *major iteration* to be the operations performed between returns to Step 1 in both algorithms. *Simplex iterations* are the number of simplex algorithm pivots performed on any of the linear programs considered by the algorithms.

Proposition: Let $\{x^\nu\}$ be a sequence of points generated by the multicut algorithm and let $\{y^\nu\}$ be a sequence generated by the L–shaped algorithm. Then, if at all major iterations, x^ν and y^ν belong to the same cells of the decomposition of $\mathcal{Q}$, the number of major iterations needed by the multicut algorithm will be less than or equal to the number of major iterations of the L–shaped algorithm.

Proof: If the conditions of the propostion are met, then constraints (5) and (12) are the same in the L–shaped and multicut algorithms. Each constraint in (6) corresponds to K constraints in (13) such that $E_l = \sum_{k=1}^{K} E_{l(k)}$ and $e_l = \sum_{k=1}^{K} e_{l(k)}$. Hence, if $(x, \theta_k), k = 1, \ldots, K$ are feasible in (11) – (13) for $k = 1, \ldots, K$, then $(x, \theta = \sum_{k=1}^{K} \theta_k)$ is feasible in (4)–(6). Therefore, the multicut algorithm objective value, $z(\text{multi}) \geq z(\text{L–shaped})$, the L–shaped algorithm objective value. If $z(\text{L–shaped}) = z^*$, the optimal value in (1), then $z(\text{multi}) = z^*$.■

Note that x^ν and y^ν belong to the same cell of the Note that x^ν and y^ν belong to the same cell of the decomposition if the recourse function $Q(x, \xi)$ is linear in ξ for each iterate x. Whenever a nonlinearity is detected, however, the iterates generally diverge. In addition, whether the iterate points belong to cells that

are close to or far from the optimal point is partly a matter of chance. Therefore, the L–shaped method can conceivably do better than the multicut approach (the reverse is obviously also true) in terms of number of major iterations. We illustrate this by the example in Appendix B. Other examples where the multicut approach does better than the L-shaped method can easily be constructed (see Appendix A).

Since none of the methods is superior to the other in all circumstances, the efficiency of the two approaches is measured in terms of worst-case analysis on the number of major iterations.

Definition:

Let $b(\xi)$ represent the maximum number of different slopes of $Q(x,\xi)$ in any direction parallel to one of the axes for a given ξ , i.e. the maximum number of different cells (of the polyhedral decomposition of $K_1 \cap K_2$ relative to $Q(x,\xi)$ for a given ξ) encountered by any ray (parallel to one of the axes) originating at a point arbitrarily chosen in $K_1 \cap K_2$.

Define $b = \max_{\xi \in \Xi} b(\xi)$ to be the "slope number of the second-stage of (2)."

For examples of $b(\xi)$ and b, see the appendices. Figure 1 illustrates the example in Appendix A. The functions Q_1, Q_2, and Q_3 represent $Q(x,\xi)$ for $\xi^1 = 1$, $\xi^2 = 2$, and $\xi^3 = 4$, respectively. In this case, $m_2 = 1$, $b(1) = b(2) = b(4) = b = 2$. In Appendix B, $m_2 = 1$, $b(\xi^1) = 2$, and $b(\xi^2) = 4 = b$.

Theorem: Let b be the slope number of the second stage of (2). Then, the maximum number of iterations for the multicut algorithm is

$$1 + K(b^{m_2} - 1) \tag{17}$$

while the maximum number of iterations for the L-shaped algorithm is

$$[1 + K(b-1)]^{m_2} \tag{18}$$

where K is the number of different realisations of ξ.

Proof: To illustrate the result, consider Figure 1 for the example in Appendix A. Let b be the slope number of the second stage of (2) ($b = 2$ in Figure 1). In the worst–case (as in Figure 1), $b(\xi) = b$ for all $\xi \in \Xi$. A single linear piece of each $Q_k, k = 1, \ldots, K$, in some direction j corresponds to a single linear piece of $\mathcal{Q}$

in direction j. Each new slope for each Q_i can result in a new slope for $\mathcal{Q}$ in direction j. For $b-1$ new slopes for each q_i, in the worst–case, $\mathcal{Q}$ obtains $1+K(b-1)$ slopes in direction j. (Note the $4 = 1+3(1)$ slopes in Figure 1.) Since this can occur in each direction $j = 1, \ldots, m_2$, $[1+K(b-1)]^{m_2}$ facets of $\mathcal{Q}$ can be generated.

In the worst–case, the L–shaped method considers every facet of $\mathcal{Q}$ (proving (18)). For the multicut approach, however, in the worst–case, one facet of each Q_k is identified in each step. Since each Q_k has at most b slopes in direction j, each Q_k has at most b^{m_2} facets. On the first iteration, a facet is identified for each $\xi_k, k = 1, \ldots, K$. Hence, the maximum number of iterations is $1+(Kb^{m_2}-K)$, proving (17).∎

The maximal number of iterations has an immediate consequence on the size of the first-stage problems to be solved. While problems of smaller size are needed in the first iterations of the L–shaped method (m_1+1 constraints, n_1+1 variables) as compared to the multicut (m_1+K constraints, n_1+K variables), the above theorem shows that the size of the problem is of the order $(b-1)^{m_2}K^{m_2}$ in the worst–case for the L–shaped approach and $K(b^{m_2}-1)$ for the multicut strategy. One can therefore expect the multicut approach to be especially efficient for problems where m_2 is large, many cuts are needed, and, as we mention in the discussion of numerical examples below, K is not larger than n_1.

In the next section, the number of facets for the particular case of simple recourse is given explicitly.

5. The Simple Recourse Case

The simple recourse problem is a particular case of the formulation (3) with non-stochastic matrix T where the function $\psi(\chi, \xi)$ is separable

$$\psi(\chi, \xi) = \sum_{i=1}^{m_2} \psi_i(\chi_i, \xi_i) \tag{19}$$

and

$$\psi_i(\chi_i, \xi_i) = \min\{q_i^+ y_i^+ + q_i^- y_i^- \mid y_i^+ - y_i^- = h_i - \chi_i, y_i^+ \geq 0, y_i^- \geq 0\}, \tag{20}$$

where

$$\xi_i = (q_i^+, q_i^-, h_i).$$

Assume that, for each i, ξ_i can take on J different values (where for simplicity of exposition J is assumed to be the same for all i).

Then, using the multicut approach consists of approximating the recourse function $\Psi(\chi)$ by the outer-linearization

$$\sum_{i=1}^{m_2}\sum_{j=1}^{J}\theta_{ij}. \tag{21}$$

Due to the simple recourse property, only two cuts of type (13) can be generated for each θ_{ij}, namely

$$\theta_{ij} \geq p_{ij}q_{ij}^{-}(\chi_i - h_{ij}) \tag{22}$$

and

$$\theta_{ij} \geq p_{ij}q_{ij}^{+}(h_{ij} - \chi_i) \tag{23}$$

where p_{ij} denotes the probability of the j^{th} realization of ξ_i.

Introducing the slack variable u_{ij} in constraint (22),

$$p_{ij}q_{ij}^{-}(\chi_i - h_{ij}) + u_{ij} = \theta_{ij} \tag{24}$$

and substituting θ_{ij} from (24) into (21)–(23), the simple recourse problem (19), (20) is equivalent to

$$\begin{array}{llllll}
\min & cx & +\sum_{i=1}^{m_2}\sum_{j=1}^{J} p_{ij}q_{ij}^{-}(\chi_i - h_{ij}) & +\sum_{i=1}^{m_2}\sum_{j=1}^{J} u_{ij} & & \\
\text{s.t.} & Ax & & & & = b \\
& Tx & -\chi & & & = 0 \\
& & & u_{ij} & & \geq p_{ij}q_{ij}(h_{ij} - \chi_i), \\
& & & & & i = 1,\ldots,m_2, j = 1,\ldots,J \\
& & & u_{ij} & & \geq 0, i = 1,\ldots,m_2, j = 1,\ldots,J,
\end{array} \tag{25}$$

where $q_{ij} = q_{ij}^{+} + q_{ij}^{-}$.

From (25), we can derive the following algorithm.

Multicut Algorithm for Simple Recourse Problems

Step 0. Set $\nu = t = 0$.

Step 1. Set $\nu = \nu + 1$. Solve the linear program

$$\begin{array}{lllll} \min & cx & +\sum_{i=1}^{m_2}\sum_{j=1}^{J} p_{ij}q_{ij}^{-}(T_i x) & +\sum_{l=1}^{t} u_l & \\ \text{s.t.} & Ax & & & = b \\ & & & u_l & \geq e_l - E_l x, l = 1, ..., t, \\ & & & u_l & \geq 0, l = 1, ..., t. \end{array} \tag{26}$$

Let (x^ν, u^ν) be an optimal solution. If $t = 0$, then u is ignored in the computation.

Step 2. For each $i = 1, \ldots, m_2$, and $j = 1, \ldots, J$, if the constraint

$$0 \geq p_{ij}q_{ij}(h_{ij} - T_i x^\nu) \tag{27}$$

is violated, define

$$E_{t+1} = p_{ij}q_{ij}T_i$$

and

$$e_{t+1} = p_{ij}q_{ij}h_{ij}$$

and set $t = t + 1$.

Note that the constraints in (26) are a subset of the constraints in (25). We use the notation u_l to represent those u_{ij} in (25) that have been identified in Step 2 of the multicut algorithm. Inequality 27 identifies any constraints in (25) that are not met on iteration ν. These constraints are added for the next iteration, $\nu + 1$.

The initial problem (27) involves m_1 constraints and n_1 variables. For this problem, the worst-case situation is when at each iteration, only one constraint (27) is violated in Step 2. Then, the maximal number of iterations is $Jm_2 + 1$. To compare with the maximal number of iterations an L-shaped algorithm

would require to solve the same problem, note that for each i, the function $\Psi_i(\chi_i) = E\psi_i(\chi_i, \xi_i)$ contains $J + 1$ facets and since $\Psi(\chi)$ is separable in i, it contains at most $(J + 1)^{m_2}$ facets. This is precisely the worst–case upper bound on the number of iterations for an L–shaped type algorithm as in the theorem of Section 4.

6. Numerical Experimentation and Conclusions

The L–shaped algorithm and the multicut method have been coded in FORTRAN in the codes NDREG and NDSEP respectively. NDREG is a two–stage version of the multi–stage code developed by Birge in [3] and described in [4]. NDSEP uses the same subroutines for linear program solutions, constraint generation and constraint elimination as NDREG. The subroutines to control where cuts are placed and to determine optimality have been modified in NDSEP to reflect the differences between the standard L–shaped method and the multicut approach.

The set of test problems and their size characteristics appear in Table 1. The first four problems are small energy examples with varying objectives and constraints and the last example is a stochastic two–stage version of one of Ho and Loute's [7] staircase problems. These examples were chosen because of their applicability and the facet structure of their recourse functions.

The problems were solved using the FORTRAN-G compiler at The University of Michigan on an Amdahl 5860. The number of major iterations, simplex iterations and CPU seconds are given for each problem in Table 2, where "Single Cut" refers to NDREG and "Multiple Cuts" refers to NDSEP. Both NDREG and NDSEP used the bunching approach (Wets [14]) for solving second–period problems. They also both included the deletion of slack cuts which resulted in savings of up to twenty percent in CPU times.

The results in Table 2 illustrate the effectiveness of the multicut approach and some of its shortcomings. In each example, the number of major iterations is reduced. This is due to the passing of more information on each major iteration as noted above. A difficulty arises, however, because of the increased size of (11) - (13) over (4) - (6). Although (4) - (6) in the worst–case may have many more constraints than (11) - (13), program (11) - (13) is initially larger and, hence, requires more time to solve. This leads to the increased

time in solving NRG4 by NDSEP. NDSEP, in fact, spends 2.8 more CPUs solving (11) - (13) than NDREG spends solving (4) - (6) on NRG4. This problem is an especially bad case because the original problem is so small that the addition of 27 extra constraints increases its size nine-fold and has a significant slowing effect.

These examples suggest that the multicut approach can lead to significant reductions in the number of major iterations. As indicated above, the worst–case advantage of the multicut approach in limiting major iterations is enhanced as m_2 increases in size. The experiments show that the multicut approach is most effective when the number of realizations K is not significantly larger than the number of first period constraints n_1. When K is large relative to n_1, it may be advantageous to use a hybrid approach in which subsets of the realizations are grouped together to form a reduced number of combination cuts. The worth of this and other strategies is, however, problem dependent and should be demonstrated through experimentation in different and varied application areas.

Table 1. Problem parameters

Problem	Period 1 (A)			Period 2 (W)			Realizations
	n_1	m_1	ρ^*	n_2	m_2	ρ^*	K
NRG1	7	3	1.000	20	8	0.375	3
NRG2	7	3	1.000	20	8	0.375	3
NRG3	7	3	1.000	20	8	0.375	9
NRG4	7	3	1.000	20	8	0.375	27
SCAGR7.S2	36	16	0.191	79	39	0.092	8
*fraction of elements (excluding slack variable elements) which are nonzero							

Table 2. Experimental results

Problem	Single Cut			Multiple Cuts		
	Major Iterations	Simplex Iterations	CPUs	Major Iterations	Simplex Iterations	CPUs
NRG1	10	117	0.34	6	64	0.23
NRG2	13	163	0.49	9	92	0.35
NRG3	14	196	1.26	8	121	1.11
NRG4	14	207	3.19	7	166	5.66
SCAGR7.S2	10	138	1.66	7	108	1.40

Appendix A

Assume that $n_1 = 1$, $n_2 = 2$, $m_2 = 1$, $K = 3$, $W = (1 - 1)$, and $Q(x, \xi) = \{\xi - x$, if $x \leq \xi$; $x - \xi$, if $x \geq \xi\}$, and that ξ can take on the values 1, 2, and 4, each with probability 1/3. Assume also $cx = 0$ and $0 \leq x \leq 10$. Note that Q has two slopes for each ξ, hence, $b = 2$.

Figure 1 represents the functions $Q_1(x)$, $Q_2(x)$, $Q_3(x)$, and $\mathcal{Q}(x)$. Since the first-stage objective cx is zero, $\mathcal{Q}(x)$ is also the function $z(x)$ to be minimized. Assume the starting point is $x^1 = 0$. The sequence of iterations for the L-shaped method would be:

Iteration 1:

x^1 is not optimal; send the cut

$$\theta \geq 7/3 - x.$$

Iteration 2:

$x^2 = 10, \theta^2 = -23/3$ is not optimal; send the cut

$$\theta \geq x - 7/3.$$

Iteration 3:

$x^3 = 7/3$, $\theta^3 = 0$ is not optimal; send the cut

$$\theta \geq \frac{x+1}{3}.$$

Iteration 4:

$x^4 = 1.5$, $\theta^4 = 2.5/3$ is not optimal; send the cut

$$\theta \geq \frac{5-x}{3}.$$

Iteration 5:

$x^5 = 2$, $\theta^5 = 1$, which is the optimal solution.

Starting from $x^1 = 0$, the multicut approach would yield the following sequences:

Iteration 1: x^1 is not optimal; send the cuts $\theta_1 \geq \frac{4-x}{3}$; $\theta_2 \geq \frac{2-x}{3}$, $\theta_3 \geq \frac{1-x}{3}$.

Iteration 2: $x^3 = 10$, $\theta_1^2 = -2$, $\theta_2^2 = -8/3$, $\theta_3^2 = -3$ is not optimal;

send the cuts $\theta_1 \geq \frac{x-4}{3}$, $\theta_2 \geq \frac{x-2}{3}$, $\theta_3 \geq \frac{x-1}{3}$.

Iteration 3: $x^3 = 2$, $\theta_1^3 = 2/3$, $\theta_2^3 = 0$, $\theta_3^3 = 1/3$ is the optimal solution.

Therefore, by sending separate cuts on $Q_1(x)$, $Q_2(x)$, and $Q_3(x)$, the full description of $Q(x)$ is obtained in two iterations.

Appendix B

Assume $n_1 = 1$, $m_2 = 3$, $n_2 = 6$,

$$W = \begin{pmatrix} 1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

and $K = 2$ realizations of ξ with equal probability 1/2. These realizations are $\xi^1 = (q^1, h^1, T^1)$ and $\xi^2 = (q^2, h^2, T^2)$, where $q^1 = (1,0,0,0,0,0)$, $q^2 = (3/2,0,2/7,1,0,0)$, $h^1 = (-1,2,7)^T$, $h^2 = (0,2,7)^T$, and $T^1 = T^2 = (1)$. For the first value of ξ, $Q(x,\xi)$ has two pieces, such that

$$Q_1(x) = \begin{cases} -x-1 & \text{if } x \le -1. \\ 0 & \text{if } x \ge -1. \end{cases}$$

For the second value of ξ, $Q(x,\xi)$ has four pieces such that

$$Q_2(x) = \begin{cases} -1.5x & \text{if } x \le 0, \\ 0 & \text{if } 0 \le x \le 2, \\ 2/7(x-2) & \text{if } 2 \le x \le 9, \\ x-7 & \text{if } x \ge 9. \end{cases}$$

Assume also that x is bounded by $-20 \le x \le 20$ and $c = 0$. Starting from any initial point $x^1 \le -1$, one obtains the following sequence of iterate points and cuts for the L-shaped method. Note that four slopes occur for Q_2 , hence, $b = 4$.

Iteration 1: $x^1 = -2$, θ^1 is omitted; new cut $\theta \ge -0.5 - 1.25x$.

Iteration 2: $x^2 = +20$, $\theta^2 = -25.5$; new cut $\theta \ge 0.5x - 3.5$.

Iteration 3: $x^3 = 12/7$, $\theta^3 = -37/14$; new cut $\theta \ge 0$.

Iteration 4: $x^4 \in [-2/5, 7]$, $\theta^4 = 0$.

If x^4 is chosen to be any value in [0,2] then the algorithm terminates at Iteration 4.

The multicut approach would generate the following sequence.

Iteration 1: $x^1 = -2$, θ_1^1 and θ_2^1 omitted; new cuts $\theta_1 \ge -0.5x - 0.5$, $\theta_2 \ge -3/4x$.

Iteration 2: $x^2 = 20$, $\theta_1^2 = -10.5$, $\theta_2^2 = -15$; new cuts $\theta_1 \ge 0$, $\theta_2 \ge 0.5x - 3.5$.

Iteration 3: $x^3 = 2.8$, $\theta_1^3 = 0$, $\theta_1^3 = -2.1$; new cut $\theta_2 \ge 2/7\ (x-2)$.

Iteration 4: $x^4 = 0.552$, $\theta_1^4 = 0$, $\theta_2^4 = -0.414$; new cut $\theta_2 \ge 0$.

Iteration 5: $x^5 = 0$, $\theta_1^5 = \theta_2^5 = 0$, STOP.

References

[1] Benders, J.F., "Partitioning procedures for solving mixed-variables programming problems," **Numerische Mathematik 4**(1962) 238-252.

[2] J.R. Birge, "A Dantsig-Wolfe decomposition variant equivalent to basis factorisation," **Mathematical Programming Study 24**(1985) 43-64.

[3] J.R. Birge, "Decomposition and partitioning methods for multistage stochastic linear programs", **Operations Research 33**(1985)989-1007.

[4] J.R. Birge, "An L-shaped method computer code for multi-stage stochastic linear programs", in: Y. Ermoliev and R. J-B. Wets, eds., **Numerical methods in stochastic programming**, to appear.

[5] S.J. Garstka and D.P. Rutenberg, "Computation in discrete stochastic programs with recourse", **Operations Research 21**(1973) 112-122.

[6] S.J. Garstka and R. J-B. Wets, "On decision rules in stochastic programming", **Mathematical Programming 7**(1974) 117-143.

[7] J.K. Ho and E. Loute, "A set of staircase linear programming test problems", **Mathematical Programming 20** (1981) 245-250.

[8] P. Kall, "Computational methods for solving two-stage stochastic linear programming problems", **Zeitschrift fuer angewandte Mathematik und Physik 30**(1979) 261-271.

[9] L.S. Lasdon, **Optimization Theory for Large Systems**, Macmillan, New York, 1970.

[10] F.V. Louveaux, "A solution method for multistage stochastic programs with application to an energy investment problem", **Operations Research 28**(1980) 889-902.

[11] J. Silverman, "An accelerated nested decomposition algorithm for the solution of multi-period multi-scenario LPs", unpublished notes, Department of Operations Research, Stanford University (Stanford, CA, 1983).

[12] R. Van Slyke and R. J-B. Wets, "L-shaped linear programs with application to optimal control and stochastic programming", **SIAM Journal on Applied Mathematics 17**(1969)638-663.

[13] R. J-B. Wets, "Characterisation theorems for stochastic programs", **Mathematical Programming** 2(1972) 166-175.

[14] R. J-B. Wets, "Stochastic programming: solution tec' niques and approximation schemes", in A. Bachem, M. Groetschel, B. Korte, eds., **Mathematical programming: state-of-the-art 1982** (Springer-Verlag, Berlin, 1983, pp. 506–603).

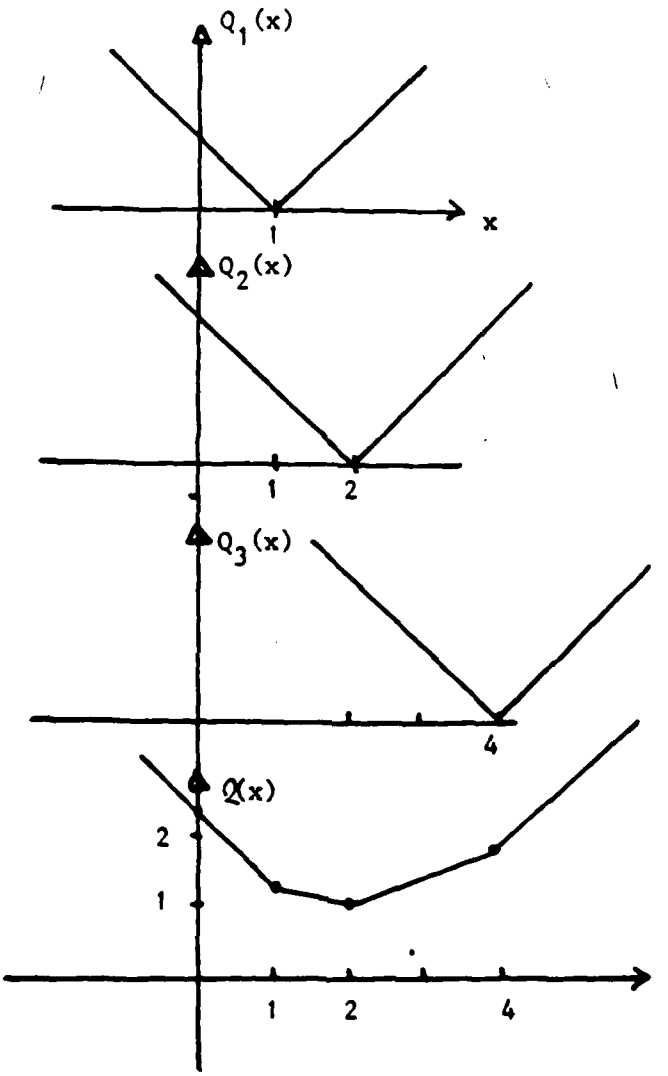


Figure 1. Recourse Functions

END

12-87

DTIC